CodeVisionAVR Revision History

V1.24.7e Commercial Release

- enhanced the peephole optimizer
- increased the maximum name length of an identifier to 64 characters
- fixed: in the *Medium* and *Large* memory models only, the RAMPZ register was not set correctly when accessing, using a constant index, the contents of an array stored in FLASH memory
- fixed: casting a variable placed in a register pair to a pointer to function produced assembly errors
- fixed: source line numbering was not correct when printing source lines that were spanned across several lines
- added the File|Print Preview feature
- fixed: optimization level information wasn't saved in the project file during project configuration, so it always defaulted to Maximal when this file was reloaded
- fixed: setting for ATtiny2313: Timer0, Mode:Ph. correct PWM top=OCR0A, OutputA: Disconnect, OutputB: Non-inverted PWM produced the *Cannot focus a disabled or invisible window* error message in the CodeWizardAVR
- fixed: no code was created in CodeWizardAVR for the Timer1 Compare A match interrupt service routine for the ATtiny25, ATtiny45 and ATtiny85 chips
- corrected the names of the SPI registers in *spi.lib* for the ATmega164, ATmega324 and ATmega644 chips
- corrected the names of the SPI registers in CodeWizardAVR generated code for the ATmega164, ATmega324 and ATmega644 chips
- in the CodeWizardAVR renamed the DS1302 /RST pin as CE in order to comply with the newer MAXIM/Dallas Semiconductor datasheets
- added the *unsigned char rtc_read(unsigned char address)* and *void rtc_write(unsigned char address, unsigned char data)* functions to the DS1307 RTC support library (ds1307.h & ds1307.lib). Many thanks to Mr. Yuri G. Salov for the DS1307 library improvement.
- increased the maximum ADC clock frequency that may be specified in CodeWizardAVR to 250kHz

V1.24.7d Commercial Release

- improved error reporting for assembly errors
- changed the declaration from *char strpos(char *str, char c)* to *signed char strpos(char *str, char c)* in the string.h header file
- changed the name of the SPMCR register to SPMCSR in the 90pwm2.h and 90pwm3.h header files
- the optional User Program is now executed after a successful Make even if the user cancels chip programming

V1.24.7c Commercial Release

- added in Settings|Programmer the possibility to specify a baud rate of 115200 for communication with a modified AVR910 programmer

V1.24.7b Commercial Release

- fixed: “pointless integer comparison with an out of range constant” warning was generated in certain cases when the result of the sizeof operator was compared with a constant
- added the possibility to disable the above mentioned warning in Project|Configure|CCompiler|Messages
- fixed: if there were no local variables declared in a function, then it’s parameters were “out of scope” when trying to Watch them in AVR Studio
- fixed: an error message that the size of the constants exceeds 65536 bytes was issued when compiling boot loaders for chips with 128kbytes or more FLASH memory
- fixed: no debugging information was generated in the COFF object files for code located in the .asm file at line numbers exceeding 65535
- the assembly file created by the compiler is now assembled with Atmel’s AVRASM2 assembler V2.1.0 distributed with AVR Studio 4.12
- added the definitions for the EXT_INT0, EXT_INT1, EXT_INT2 interrupt vectors in the mega164.h, mega324.h and mega644.h header files, in order to make them consistent with code generated by CodeWizardAVR
- added the definition for the TCCR0B Timer 0 control register in the mega164.h, mega324.h and mega644.h header files
- fixed: the CodeWizardAVR generated an incorrect TWI interrupt vector name for the ATmega 48/88/168 and ATmega406 chips

V1.24.7a Commercial Release

- improved the peephole code optimizer
- improved the common sub-expression elimination optimizer
- added warnings for pointless comparisons of integer variables with out of range constants
- fixed: the CodeWizardAVR generated incorrect initialization code for the EIMSK register of the ATmega325 and ATmega645 chips for pin change interrupts

V1.24.7 Commercial Release

- improved the peephole code optimizer
- added the common sub-expression elimination optimizer
- added the Medium, respectively Large, memory models which allow pointers to address the entire FLASH memory for chips with 128kbytes, respectively 256kbytes or more, of FLASH
- added support for ATmega164, ATmega324 and ATmega644 chips
- added a warning for cases when the register keyword is used, but the variable can’t be allocated to a register
- significantly improved the speed of the isqrt and lsqrt functions from math.h
- modified the software bit-banged I/O I2C functions ‘i2c_read’ and ‘i2c_write’ (i2c.h) to test if the I2C slave is holding the SCL line low in order to extend the read or write cycles
- documented the predefined macro __BUILD__ which specifies the build number
- added the possibility to specify the optimization level in Project|Configure|C Compiler|Code Generation

- added the #warning preprocessor directive
- fixed: when ANSI char to int promotion compilation option was turned on and the ++ or -- operators were applied to a struct member of char type used as an array index, the next struct member had it's LSB value trashed
- fixed: the post-increment and post-decrement operators have now higher precedence than the cast operator
- fixed: the CodeWizardAVR allowed PORTF and PORTG of the ATmega64/128 to be used for 1 Wire and I2C functions, which led to assembly errors, as these ports were located in extended I/O space in RAM
- fixed: the CodeWizardAVR generated incorrect initialization of the ATtiny2313 MCUCR register for external interrupts 0 and 1 triggered by any change, falling or rising edge
- fixed: no warning was issued for global variables that were declared, but not used
- fixed: the WDTCSR register was incorrectly named WDTCR in the mega640.h, mega1280.h, mega1281.h, mega2560.h and mega2561.h header files
- added the possibility to search and replace inside selected blocks of text for Edit|Find and Edit|Replace menu commands
- added the possibility to confirm a replacement for the Edit|Replace menu command
- added the Edit|Replace in Files menu command

V1.24.6 Commercial Release

- improved the code optimizer
- improved the error/warning reporting
- added support for ATmega406 chip in the Compiler and CodeWizardAVR
- added support for the ATmega1280/1281/2560/2561/640 chips in the Compiler and Programmer
- added the 'poweroff' function in the Power Management functions (sleep.h)
- added support for the Maxim DS18B20 temperature sensor (ds18b20.h and ds18b20.lib)
- fixed: the initialization sequence for the IVSEL and IVCE bits from the MCUCR register when compiling bootloaders for the ATmega165/168/169/48/325/3250/329/3290/645/6450/649/6490/88 chips
- CodeVisionAVR no longer needs to be installed in a path without embedded spaces under Windows NT 4.0, 2000 or XP
- improved the Programmer's behaviour when switching between Windows applications
- fixed: the Programmer failed to program the AT90S4414/8515 FLASH memory when used with STK500/AVRISP firmware V2.0 introduced in AVR Studio 4.11
- fixed: the Programmer failed to program the FLASH memory for the ATmega168, AT90PWM2 and AT90PWM3 chips
- fixed: the CKOUT and CKDIV8 fuse bits were swaped when programming the ATtiny25/45/85 chips
- added the possibility to enable/disable the automatic collapsible blocks in Settings|Editor

V1.24.5 Commercial Release

- improved the code optimizer
- added support for ATmega325/3250, ATmega329/3290, ATmega645/6450, ATmega649/6490 chips in the Compiler, CodeWizardAVR and Programmer
- added support for AT90PWM2 and AT90PWM3 chips in the Compiler and Programmer
- added initialization of the TWPS0 and TWPS1 bits of the TWSR register in the code generated by the CodeWizardAVR for the TWI controller
- added support for the new STK500/AVRISP communication protocol (firmware 2.0) introduced in AVR Studio 4.11
- added the possibility to specify the color of the collapsible blocks bar in Settings|Editor

V1.24.4a Commercial Release

- improved the code optimizer
- improved ANSI C compatibility
- added the “possible loss of precision” warning when the result of an assignment is truncated
- added the “constant out of range” warning when a constant is truncated during an assignment
- added the possibility to enable/disable word alignment for members of structures located in FLASH (Project|Configure|C Compiler|Code Generation|Word Align FLASH Struct). This option should be unchecked for new projects in order to reduce code size.
- fixed: when a pointer to a structure or union was casted to a pointer to char, int, long or float, the size of the pointed data type remained the size of the structure or union
- fixed: when a constant was casted to a pointer, the size of the pointed data type was 1 no matter of the size of the pointed data type
- fixed: casting a constant involved in a mathematical operation with another constant was applied to the constant result of the operation, not to the operand itself
- fixed: if in a switch statement a case label 0 was used after a default label then the compiler generated a “duplicate case label 0” error
- fixed: missing XDIV register initilialization for ATmega64/128 in CodeWizardAVR generated code
- corrected the behavior of the Clipboard History window
- added the possibility to edit the name and path of a file that is part of the project (Project|Configure|Files)
- added collapsible blocks for comments and definitions
- Undo and Redo now scroll the editor view to the place where the change was done
- added the possibility de define custom color for syntax highlighting in Settings|Editor

V1.24.4 Commercial Release

- improved code optimizer
- significantly improved the compile speed for large projects
- improved ANSI compatibility
- improved register saving/restoring in interrupt service routines that don’t alter register R30
- added support for static and a maximum of 8 local bit variables
- no code is generated for C functions that contain inline assembly code and are not used in the program
- added support for the ATtiny25/45/85 and ATmega165 chips in the Compiler, CodeWizardAVR and Programmer
- fixed: disabled the usage of the hardware multiplier for ATtiny2313, as this chip doesn’t have one
- fixed assembly errors when global bit variables were located in GPIOR registers with addresses higher than 0x1F
- added the possibility to individually enable/disable warnings using the Project|Configure|C Compiler|Messages menu
- corrected the names of the MCUSR and ADCSRB registers in CodeWizardAVR generated code for AT90CAN128
- fixed the CAN controller Baud Rate initialization for AT90CAN128 in the CodeWizardAVR
- corrected the name of the MCUSR register in CodeWizardAVR generated code for ATmega48/88/168
- added the watchdog timeout interrupt for ATmega48/88/168 in the CodeWizardAVR
- added the definition for the PCMSK2 I/O register in the mega48.h, mega88.h and mega168.h header files
- added the possibility, in the Programmer, to copy the Lock and Fuse Bits read values to the corresponding programming settings
- for STK500 and AVRISP the Programmer now adjust the correct duration of the SCK pulse based on the clock frequency of the chip to be programmed, allowing safer in-system programming for chips running at low clock

frequencies
- added ATmega88 and ATmega168 programming support for AVRISP and STK500 (requires AVRISP/STK500 firmware update to V1.18 using AVR Studio 4.10)
- the Programmer and Terminal may now share the same COM port
- the Navigator now displays the RAM placement address or register allocation for global variables
- improved the syntax highlighting in the editor
- added the possibility to do syntax highlighting for user defined keywords (Settings|Editor)
- added the possibility to comment/uncomment a block of text using Ctrl+[and Ctrl+]
- added the possibility to hide the Find in Files tab by right clicking with the mouse on it
- added a Code Templates window from which the user can drag and drop code sequences to the editor window
- added a Clipboard History viewer window
- significantly reduced the time needed to close large projects
- fixed: the IDE sometimes crashed when a file opened in the CodeVisionAVR's editor was simultaneously edited with an external editor

V1.24.3b Commercial Release

- increased the maximum value of the Delay Multiplier (for parallel port programmers in the Settings|Programmer menu) to 99

V1.24.3a Commercial Release

- fixed: in bootloaders the unused interrupt vectors now point to the start of the bootloader, instead of address 0
- added 16 bit access to I/O register ICR1 in the mega8.h header file
- improved the functionality of the Delay Multiplier for parallel port programmers (Settings|Programmer menu)

V1.24.3 Commercial Release

- added pointers to pointers support
- improved the peephole optimizer
- added AT90CAN128 support
- modified signed division result truncation to comply with the C99 standard
- added support for simplified pointers to functions syntax
- added the io.h file with the I/O register definitions for ALL the AVR chips
- ATtiny2313 now supports the double baud rate in the CodeWizardAVR
- enabled the DWEN fuse programming for the ATtiny2313
- added the possibility to specify if the fuse bits will be programmed After Make or during Program All
- added the possibility to configure the Terminal's font, number of rows and columns
- improved the Find in Files dialog
- minor improvements in the IDE

V1.24.2c Commercial Release

- improved the peephole optimizer

V1.24.2b Commercial Release

- improved the peephole optimizer for the ++ and -- operators applied to global pointers to char and int, allocated to registers in the tiny memory model
- fixed: error message when applying the ++ and -- operators to a global pointer allocated to a register
- fixed: a small bug in the C preprocessor which led to errors in handling certain literal strings which contained the \\ and \" escape sequences one after the other (example: "\\\"")
- increased the height of the Information window, so the text will fit in it on some systems
- added the possibility to change the project's file compilation order in Project|Configure|Files
- added the possibility to specify buffer sizes larger than 256 for the buffered interrupt driven serial communication code generated by the CodeWizardAVR

V1.24.2a Commercial Release

- fixed: the PORT I/O register addresses are now correct for inline assembly code generated by the CodeWizardAVR for the ATmega169 chip
- fixed: the Terminal window remaining opened in certain situations

V1.24.2 Commercial Release

- improved the peephole optimizer
- the register allocator will now also allocate global pointers to registers. **The allocation of global variables to particular registers may change compared to previous versions of the compiler.**
- improved the register allocator for local variables
- added the Project|Syntax Check function in the IDE
- improved the Edit|Find in Files functionality in the IDE
- added the Edit|Find Previous (Ctrl+F3) option in the IDE
- added STK500/AVRISP support for ATmega48 in the Programmer
- fixed in the CodeWizardAVR: no code was generated for Timer 3 initialization for the ATmega162 chip

V1.24.1e Commercial Release

- improved the watchdog disabling in the startup code generated by the compiler for the ATmega48/88/168 chips
- improved the watchdog initialization in the code generated by the CodeWizardAVR for the ATmega48/88/168 chips
- improved the Find, Find in Files and Replace dialogs used in the IDE

V1.24.1d Commercial Release

- improved the code optimization in interrupt services routines
- fixed assembly errors that appeared in some cases when the LSB of a local pointer was allocated to an odd numbered register and the MOVW enhanced core instruction was generated
- improved the C preprocessor

V1.24.1c Commercial Release

- improved ANSI compatibility: the const keyword, used when declaring pointer function arguments, now specifies that these arguments may not be modified by the function. For pointers to FLASH arguments, the flash keyword must be now used instead of const

V1.24.1b Commercial Release

- improved the common block packing optimizer for interrupt service routines
- fixed a bug in “smart” register saving/restoring in interrupt service routines, which led to data stack unbalance when several return statements were present
- improved error reporting when declaring global bit variables
- added the possibility to suppress assembly macro expansion in the .lst file (Settings|Assembler|Expand Macros in .lst file)
- corrected in the CodeWizardAVR generated code the names of the ICR3L and ICR3H registers

V1.24.1a Commercial Release

- fixed a bug in the peephole optimizer which could lead to assembly errors if optimization for Size was specified in Project|Configure|C Compiler|Code Generation

V1.24.1 Commercial Release

- added support for “smart” saving/restoring of affected registers in interrupt service routines. The #pragma savereg directive is not needed anymore
- improved the peephole optimizer
- added the constant literal strings merging optimizer
- improved the register allocator, registers are now allocated for local pointers too
- improved the dead code optimizer
- improved error and warning reporting
- modified the startup initialization code for watchdog disabling and global variable initialization. Projects that use an external startup.asm file must have this file updated with the new one supplied with the compiler
- improved ANSI C compatibility
- fixed a rare bug which occured when testing a bit in SREG (if (SREG.0) ... didn’t work, but if (SREG.0==1)... did)
- in the header file mega32.h, the TWSI interrupt vector was renamed to TWI
- modified the CodeWizardAVR so that it will not generate code with #pragma savereg for interrupt service routines
- added initialization for ICR1 and ICR3 in the CodeWizardAVR
- fixed in the CodeWizardAVR: removed ADC auto-trigger source for ATmega8(L)
- fixed in the CodeWizardAVR: the ADATE bit in ADCSRA will be 0 for ADC free running mode
- fixed AT86RF401 parallel port (STK200) FLASH programming
- fixed ATmega64(L) STK500/AVRISP programming
- improved the Programmer menus
- added one-button programming
- various small fixes and improvements in the IDE

V1.24.0 Commercial Release

- added the malloc, calloc, realloc and free memory allocation standard library functions (stdlib.h)
- added the _HEAP_START_ and _HEAP_SIZE_ predefined macros
- improved the constant folding for calculations employing the ? : operator
- updated the LCD libraries for better compatibility
- the lcd_init function in the LCD libraries now detects the presence of a LCD module, returning 1 on success and 0 on failure
- updated the tiny2313.h header files with the definitions for the PORTA, DDRA and PINA I/O registers
- fixed the ## preprocessor operator not working in nested macros
- added support for the ATmega88 and ATmega168 chips in the Compiler, CodeWizardAVR and Programmer
- added the PORTA pins in the CodeWizardAVR for the ATtiny2313 chip
- added support for the ATmega48 chip in the CodeWizardAVR
- added support for the ATtiny2313 chip in the Programmer for STK500 and AVRISP (the firmware update from AVR Studio V4.08 is needed for these programmers)

V1.23.9b Commercial Release

- added support for the ATmega48 chip in the Compiler and Programmer
- updated the CodeWizardAVR to comply with the latest ATmega8 data sheet

V1.23.9a Commercial Release

- modified the code generator so that the global bit variables will be first allocated to GPIOR registers, if present in the chip, and then to R2..R14
- added support for the ATtiny2313 chip in the Compiler, Programmer and CodeWizardAVR
- modified the startup code for the ATtiny13 chip so that the watchdog timer could be disabled even if the WDTON fuse is programmed

V1.23.9 Commercial Release

- enhanced the peephole code optimizer
- added support for generating extended COFF object files, which allow structure and union variables watching in AVR Studio V4.07 or later
- fixed the ++ and -- operators not working on the first bit field of a structure or union
- corrected the spi function from spi.h to work with the ATmega169 chips
- the EEPROM location at address 0 is now used by the compiler (for newly created projects), as the corruption of this location was fixed by Atmel
- improved the code generated by the CodeWizardAVR for buffered interrupt driven communication which uses parity
- fixed the values displayed by the CodeWizardAVR in the External SRAM Page Configuration list box
- fixed the wrong error message issued by the CodeWizardAVR when a LCD was used on PORTC on an ATmega128(L) and the Two Wire Interface was enabled
- added support for the ATtiny13 chip in the Programmer
- added support for the in-system programmer described in the Atmel application note AVR910

V1.23.8d Commercial Release

- added support for the ATtiny13 chip in the Compiler and CodeWizardAVR
- added support for reading long integers in scanf and sscanf functions from stdio.h
- added the vprintf and vsprintf functions in stdio.h
- typedef now allows defining the pointers to functions data type
- enhanced the peephole code optimizer
- improved error and warning reporting
- fixed assembly errors when calling the power management functions for the ATmega162 and ATmega8515 chips
- the ++ and -- operators now work correctly when applied to bit fields in structures or unions
- any expression can now be used when initializing bit fields in structures or unions
- fixed a bug that occurred when subtracting two pointers that pointed to structures or unions
- fixed: the sizeof operator returned incorrect size for arrays of structures or unions located in FLASH or EEPROM
- when using external RAM and external interrupts, the CodeWizardAVR will now correctly set the MCUCR and EMCUCR registers for the ATmega162 chip
- the Timer 1 operating modes selection list box for the ATmega32 chip is now correct in the CodeWizardAVR
- in the CodeWizardAVR the external interrupts 0 and 1 now allow also the “any level” setting for the AT90S2333 and AT90S4433 chips
- fixed in CodeWizardAVR: the SPI DO pin for ATtiny26 is now PORTB bit 1
- fixed: incorrect FLASH programming for the ATmega8535 chip using parallel port programmers

V1.23.8c Commercial Release

- added the possibility to stop the compilation process by pressing the Stop Compilation button or using the Project| Stop Compilation menu command
- the dimension of an array of structures can be now omitted during its initialization
- fixed the incorrect amount of copied data, when assigning an element of an array of structures to another structure
- changed the name of the ADCSR register to ADCSRA in the mega32.h header file, in order to comply with the latest datasheet

V1.23.8b Commercial Release

- header .h files, that are present in the Project|Configure|Files list, are not compiled/linked anymore if they have been previously #included in one of the project’s program modules
- in the Project|Configure|C Compiler|Code Generation, and in the CodeWizardAVR, the AVR clock frequency can be now specified up to 40MHz, as required by the new FPSLIC chips
- fixed in the CodeWizardAVR: for the ATmega169 the name of the MCUSR register is now correctly used when generating code for reset source identification
- fixed in the CodeWizardAVR: for the ATmega64/128 the TCCR2 register is now correctly initialized when specifying Timer/Counter 2 clock source as pulses on pin T2

V1.23.8a Commercial Release

- added support for the ATmega162 chip in the Compiler, CodeWizardAVR and the Programmer
- the lcd_putsf function from lcd.h, lcd4x40.h and lcdstk.h can now be used in boot loaders for the ATmega128 chip
- fixed a bug resulting in incorrect assigning of a structure located in EEPROM to a structure in RAM
- corrected the Fuse Bits settings for ATmega161 according to the latest data sheet

- removed the warning message issued by the Programmer about the necessity to upgrade the firmware of the STK500/AVRISP
- the Programmer now supports reading the STK500/AVRISP firmware version in the Read|Programmer's Firmware Version menu

V1.23.8 Commercial Release

- added support for bit fields in structures and unions
- enhanced the code optimizer
- added automatic .hex file generation after Make, at the same time with .cof, .rom and .eep
- the atof function (stdlib.h) now accepts spaces and the + sign before the number to convert
- added support for the ATmega8535, AT76C711 and AT86RF401 chips in the Compiler
- corrected the interrupt vector numbers for ANA_COMP and ADC_INT in mega169.h
- updated the interrupt vectors definitions in mega32.h
- updated lcd.lib, lcd4x40.lib and lcdstk.lib to work with AVR chips up to 16MHz
- the Information window now also displays the FLASH code size usage in %
- the Project Navigator now also displays the #included files
- added support for ATmega8535 and AT86RF401 in the CodeWizardAVR
- corrected, as per the new datasheets, the prescaler values of Timer 0 for the ATmega16/32/323 in the CodeWizardAVR
- corrected, as per the new datasheets, the prescaler values of Timer 2 for the ATmega128 in the CodeWizardAVR
- made correction in the CodeWizard so that the bit 7 of UCSRC will be set during USART initialization for ATmega16/32/323/8/8515
- added support for ATmega8535 and AT86RF401 in the Programmer
- the Programmer can now read all the oscillator calibration bytes for ATtiny26 and ATmega64/128
- corrected fuse bits names for ATmega32 in the Programmer

V1.23.7 Commercial Release

- the printf and sprinft Standard I/O functions now support printing floats (%f, %e, %E) with width and precision. To enable printing floats, select Project|Configure|C Compiler|(s)printf features: float, width, precision.
- the printf, sprintf, scanf and sscanf Standard I/O functions are now reentrant
- libraries can be nested now
- additional paths for #include and library files can now be specified in Project|Configure|C Compiler|Paths
- storing pointers in FLASH and EEPROM now use the following syntax: *[<type storage modifier>] type * [<pointer storage modifier>] pointer_name;* or *type [<type storage modifier>] * [<pointer storage modifier>] pointer_name;*
- a function that is pointed by a pointer to function, pointer that is used as an argument of another function, can now also have arguments
- added the possibility to merge an additional .ROM file when programming the chip in Project|Configure|After Make. This is very useful for merging an application, that was previously compiled, with a boot loader
- added support for the new ATmega169 and ATmega8515 chips in the Compiler, CodeWizardAVR and Programmer
- pressing Ctrl+Del in the Editor window now deletes all the characters in a word, that are located to the right of the cursor
- added the Program Preview option in CodeWizardAVR
- improved the code generated by the CodeWizardAVR for interrupt driven serial communication

- fixed a bug which generated assembly errors when using large local arrays
- fixed several bugs in programming the ATtiny26 chip
- fixed a bug which resulted in incorrect error messages when programming chips which have the RSTDISBL fuse

V1.23.6b Commercial Release

- improved the code optimizer for the ATmega chips that use the enhanced core instructions
- fixed the definition of the ds2430_read_appreg_block function in ds2430.lib
- fixed a bug in the CodeWizardAVR generated code for the ADC scanning multiple inputs (replaced ... |adc_index with ... +adc_index)
- fixed a bug in the CodeWizardAVR that resulted in incorrect initialization of the TIMSK register for Timer/Counter0 overflow interrupt for the ATtiny26 chip
- fixed a bug in the CodeWizardAVR that resulted in setting the bit 7 of the UCSR1C register, when using interrupt driven serial communication through USART1, for the ATmega64/128 chips

V1.23.6 Commercial Release

- enhanced the dead code removing optimizer
- added support for the ATtiny26 chip in the Compiler, CodeWizardAVR and Programmer
- the _CODEVISIONAVR_ predefined macro now returns the version of the compiler as an integer. Example: for V1.23.6 it's 1236
- fixed a bug in the CodeWizardAVR which resulted in incorrect setting of the ISCxx bits in the MCUCR register for the ATmega161 chip
- in the CodeWizardAVR the TWI bit rate is now limited to 400kHz in order to comply with the I^2C bus specifications
- in the CodeWizardAVR the ADC clock frequency is now limited to 200kHz
- renamed the TWI interrupt vector for the ATmega163 chip in mega163.h to TWI in order to comply with the latest Atmel datasheets. This change is also reflected in the code generated by the CodeWizardAVR for the ATmega163 chip.
- fixed a bug in the Programmer which sometimes resulted in returning the 0xFF, 0xFF, 0xFF signature in the Read|Chip Signature menu when using the STK500 and AVRISP programmers
- small enhancements and bug fixes in the IDE

V1.23.5 Commercial Release

- added the possibility to display char strings located in FLASH in the printf and sprintf standard I/O functions. This is accomplished by using the new 'p' conversion type character. Example: printf("Hello %p","world!");
- added the possibility to debug boot loaders in AVR Studio, by enabling the Project|Configure|C Compiler|Boot Loader Debugging option. This option is available only if the Project|Configure|C Compiler|Program Type option is set to Boot Loader
- fixed a bug in the ftoa function from stdlib.h

V1.23.4 Commercial Release

- completely rewritten the printf and sprintf functions. These functions now support the precision specifier and displaying longs. Which version of these functions will be linked, can now be specified in the Project|Configure|C Compiler|(s)printf features option

- added the non-local jump functions setjmp and longjmp
- added the _MCU_CLOCK_FREQUENCY_ predefined macro that represents the AVR clock frequency in Hz, as specified in the Project|Configure|C Compiler|Clock option
- various small enhancements and fixes in the IDE

V1.23.3 Commercial Release

- improved ANSI compatibility
- added the _CHIP_ATXXXXX_ predefined macro, where ATXXXXX is the chip type in uppercase letters, selected in the Project|Configure|C Compiler|Chip option
- fixed the Find in Files not working bug

V1.23.2 Commercial Release

- improved code size and speed
- pointers can now be stored in FLASH and EEPROM
- enumerations can now be treated as being of 8 bit char data type, leading to better code size and speed. This option can be enabled from Project|Configure|C Compiler|8 bit enums or by using the new #pragma 8bit_enums+
- added the possibility to print source line numbers

V1.23.1 (new release numbering) Commercial Release

- improved ANSI compatibility of the compiler
- various small improvements/fixes in the IDE
- added the Delay Multiplier option in Settings|Programmer for the In-Systems Programmers that use the LPT port. This option is useful for programming on very fast machines.
- fixed in the CodeWizard: the F port for the ATmega64/128 chips is now bi-directional
- fixed in the CodeWizard: incorrect configuration of the DDRE, PORTE, DDRF and PORTF registers for the ATmega64/128 chips
- fixed in the CodeWizard: watchdog configuration for the ATmega8/16/64/128 chips

V1.0.2.2 Commercial Release

- removed the 64 byte limitation on the size of the local variables
- the functions can now accept arrays, structures and unions as arguments, not only pointers to them like in the previous versions
- the = operator now also works with structures and unions
- the compiler now calculates the Data Stack usage, the result being displayed in the Information window after Compile or Make
- significantly improved the speed for generating the COFF object files
- added the strtok function in String Processing Functions (string.h)
- added support for Variable Length Argument Lists (stdarg.h)
- rewritten the scanf and sscanf functions from Standard I/O Functions (stdio.h)
- added the __CODEVISIONAVR__ predefined macro
- added the possibility to #define a list of macros that are global to all the source files in the project

- removed the Global #defines option from Project|Configure|C Compiler dialog. For compatibility with old projects use the new #pragma glbdef+ at the beginning of the first source file of the project
- improved the generated code speed and size
- improved overall ANSI C compatibility of the compiler
- added support for the ATmega8/16/64/128 and AT43USB355 devices in both the compiler and the CodeWizard
- the interrupt driven buffered serial communication code generated by the CodeWizard is now written in C, not assembly like in previous versions, allowing for easier modification and understanding
- the Atmel STK500 and AVRISP are now directly controlled by the programmer, there's no need for STK500.EXE from AVR Studio anymore
- added programmer support for the Futurlec JRAVR AT90S2313 and AT90S8535 development boards
- fixed the Fuse Bits programming error for the ATmega161
- the IDE window is not maximized at startup anymore, but remembers it's previous size. This is useful for systems with large or dual monitors
- improved the licensing system. The license can be now exported on a floppy disk and reimported back on the same computer too. This is useful when doing HDD upgrades.

V1.0.2.1 Commercial Release

- added support for the ATmega8/16 and ATmega128 devices in the compiler
- updated the Power Management Functions for the ATmega128, ATmega8/16, ATmega161 and ATmega32/323
- updated the code generator in order to comply with the latest ATmega161 errata data sheet
- improved the error reporting
- improved the File|Reopen menu
- small bug fixes
- revised the Help and User Manual topics about #include, Using Interrupts and Power Management Functions

V1.0.2.0 Commercial Release

- improved the size and speed of the generated code, especially when accessing structure or union elements using pointers
- in order to improve code portability, type, structure, union and enum definitions are now visible only in the program module where they reside
- added strtok to the String Functions
- removed _strlen from the String Functions, the existing strlen function now returns unsigned int in the SMALL memory model
- fixed scanf and sscanf in the Standard C Input/Output Functions

V1.0.1.9 Commercial Release

- added support for the ATmega323 and AT94K05 FPSLIC
- added the possibility to select the type of the compiled program: Application or Boot Loader, in both the Project Configuration and the CodeWizardAVR
- added the Dallas Semiconductor DS2430 and DS2433 1 Wire EEPROMs functions
- improved the error reporting
- improved the Match Braces menu command
- fixed: the result of the sizeof operator was truncated to 8 bits

- fixed: local pointers and pointer arrays were always initialized with NULL during declaration, even if another value was used
- fixed: the Check Erasure option was always set in the AVR Chip Programmer

V1.0.1.8 Commercial Release

- improved code size and speed
- added support for the new AVRASM32 V1.52 assembler supplied with AVR Studio 3.50
- improved the Match Braces menu command
- if errors are present, the Navigator Errors and Warnings branches are now automatically expanding for the file whose Editor window has focus
- added the Gray Code Conversion Functions
- fixed the fuses programming problem for the ATmega32 device
- added a new example program how to use the ADC on the AT90S8535

V1.0.1.7 Commercial Release

- improved the code speed for devices with more than 128 bytes of SRAM
- added the _GLOBAL_DEFINES_ predefined macro
- improved the function atof in the Standard Library Functions
- added the function _strlen in the String Functions
- added to the programmer the option to preserve the EEPROM contents during chip erasure
- improved error handling for the STK500 programmer
- various small improvements in the IDE

V1.0.1.6 Commercial Release

- added support for the ATmega32 device
- added the modf function to the mathematical functions library
- improved code size
- improved the programs generated by the CodeWizardAVR for interrupt driven serial communication
- doubled the communication speed on the I2C bus
- added the _OPTIMIZE_SIZE_ and _OPTIMIZE_SPEED_ predefined macros
- added sfrw ADCW in the mega163.h header file

V1.0.1.5 Commercial Release

- improved the code size and speed of comparisons
- improved the peephole and cross-jumps optimizers
- added support for the new AVRASM32 V1.50 assembler that comes with AVR studio 3.22
- the EEPROM is now written only if the new data is different from the old one
- added the rand and srand standard library functions
- added the strstr and strstrf string functions
- rewritten the mathematical functions library and added the following new functions: ldexp, frexp, asin, acos, atan, atan2

- added the Global #define compiler option to allow macros to be globally visible in all the project files
- removed the #pragma enhcore directive and added the Enhanced Instructions compiler option to enable/disable the generation of Enhanced Core instructions for ATmega161/163 and FPSLIC devices
- added ISP support for the Atmel STK500 development board
- improved the File|Reopen menu command
- clicking in the left margin of the editor window now selects a whole line of text
- added the Ctrl+Y keyboard shortcut for deleting text lines in the editor
- changed the Redo keyboard shortcut to Shift+Ctrl+Z
- changed the Goto Line keyboard shortcut to Alt+G

V1.0.1.4 Commercial Release

- added the cross-jumping code optimization
- added the function return code optimization
- added the dead code removing optimization
- significantly improved the speed and size of the generated code for the switch operator
- added support for the ATmega161, ATmega163 and AT94K FPSLIC devices in the CodeWizardAVR
- reduced the amount of Windows system resources used by the CodeVisionAVR IDE
- replaced the 94k10.h, 94k20.h and 94k40.h header files with the 94k.h file
- added the Clear button in the Terminal

V1.0.1.3 Commercial Release

- added enum
- added multiple instruction support in #asm
- the ? operator can now have any type of operands
- if a project is opened the AVR Chip Programmer will automatically load the corresponding .rom, .eep, .hex or .bin files, if they exist
- redesigned the Terminal

V1.0.1.2 Commercial Release

- added ISP support for the Dontronics DT006 development board
- replaced all char with signed char in the header files math.h , lm75.h and ds1820.h
- improved the speed and size of the sin, cos and tan functions
- added the fmod function in math.h
- added the predefined macro _UNSIGNED_CHAR_
- added the possibility to load and save projects created in the CodeWizardAVR
- modified the lm75_init function
- added support for the Dallas Semiconductor DS1621 thermometer/thermostat
- added support for the Philips PCF8563 RTC
- enabled syntax highlighting when debugging in AVR Studio
- added the Edit|Find in Files menu command
- added the Edit|Print Selection menu command
- added the limits.h and float.h header files

- added the possibility to edit the contents of the FLASH and EEPROM buffers in the AVR Chip Programmer
- modified the keypad.c example program so it compiles with the Evaluation version

V1.0.1.1 Commercial Release

- added do while loops optimization
- optimized bit level copy operations
- added the new lcdchar.c example for defining user characters in LCDs

V1.0.1.0 Internal Release

V1.0.0.9 Commercial Release

- the project files are now automatically linked to the main file on Make, without the need to use #include
- added the new Common Block Subroutine Optimizer, that allows achieving 25% smaller code size for large programs with optimization for size turned on
- added loop, branch and subroutine call optimizations
- added macros with parameters
- added the # and ## preprocessor operators
- added the __LINE__ , __FILE__ , __DATE__ , __TIME__ predefined macros
- added the #undef , #line , #error preprocessor directives
- added the #pragma optsize compiler directive
- added the #pragma enhcore compiler directive
- local variables can be now initialized during declaration
- added functions for LCDs connected in 8 bit memory mapped mode, as used in the Kanda Systems STK200 and STK300 development boards
- added support for libraries
- added the following mathematical functions: exp, log, log10, pow, sin, cos, tan, sinh, cosh, tanh
- when invoked, the AVR Studio debugger will automatically load the .obj or .cof file
- added in CodeWizardAVR: the MISO pin is configured as an output in SPI slave mode
- added Page Setup for printing
- added the Match Braces editor command
- fixed Fuse Bit(s) programming for AT90S/LS2323, AT90S/LS2343, ATtiny12
- fixed programming of the AT90S1200
- fixed "out of scope" messages when watching local variables in AVR Studio
- added a reset button in the chip programmer
- the chip programmer buffers are now initialized with FFh
- added support for the Vogel Elektronik VTEC-ISP
- added the possibility to disable programming verification in the chip programmer
- added a chip programmer counter
- reduced the amount of Windows resources required by CodeVisionAVR
- various enhancements to the IDE
- added the new 4x4 KEYPAD example program

V1.0.0.8 Commercial Release

- added the possibility to specify the size of bit variables used by the program
- enhanced the code optimizer, code size decrease is more than 30% compared with the previous version
- added COFF file generation in order to allow variable watching in AVR Studio 3 Debugger
- added an advanced automatic register allocator
- added allocation of global char and int variables to registers up to R15
- added allocation of local char and int variables to registers from R16 to R21
- added the #pragma regalloc compiler directive
- added the register keyword
- added the typedef keyword
- added pointers to functions
- modified the >> operator to generate the ASR, respectively LSR, instructions for signed, respectively unsigned, operands
- added the short int and double data types
- added the possibility to enter constants in binary form
- added the strlenf function
- the compiler uses now only the AVRASM32 32 bit assembler, that is supplied with the AVR Studio 3 Debugger
- added support for the Dallas Semiconductor DS1302 and DS1307 real time clocks
- added support for 4x40 characters alphanumeric LCDs in the lcd4x40.h file
- programs that use buffered interrupt driven serial communication and ADC, produced by the V1.0.0.7 CodeWizardAVR, need to be generated again with the V1.0.0.8
- modified the i2c_read function in order to allow sending acknowledgments
- the i2c_init function must be called now prior to lm75_init, respectively rtc_init, functions for LM75, respectively for PCF8583 and DS1307
- increased the speed of the I2C bus functions
- renamed the ADC interrupt vector number to ADC_INT and added the ADCW declaration in the header files: 90s2333.h, 90s4433.h, 90s4434.h, 90c8534.h, 90s8535.h, mega103.h and mega603.h
- added the Navigator
- added a pop-up menu, activated with the mouse right button, in the Editor windows
- added the possibility to insert bookmarks in the edited source file
- added the possibility to visit HP InfoTech's web site directly from the IDE
- added the possibility to contact HP InfoTeche-mail by e-mail directly from the IDE
- added the possibility to run an user program after Make
- enhanced the Serial Communication Terminal
- added the possibility to save and restore the position of the Terminal window
- added support for using the ISP with Windows NT 4.0 and 2000 with User privileges
- added initialization with 0 of the ISP buffers prior to file loading
- fixed: error message when programming the BODEN fuse for the AT90S2333/4433 chip
- fixed: GPF on some errors in C source files

V1.0.0.7 Commercial Release

- added the CodeWizardAVR automatic program generator
- added multidimensional array support (up to 8 dimensions)
- added extern variables
- added to the Serial Communication Terminal the possibility to save received characters to a file and to transmit

files
- enhanced the code optimizer
- added support for the Atmel 32 bit Win32 assembler AVRASM32, supplied with the AVR Studio 3 Debugger
- added the possibility to use an external startup file
- added the possibility to transfer the license to another computer

V1.0.0.6 Commercial Release

- added floating point support
- added static variables
- added memory access functions: peekb, peekw, pokeb, pokew
- added new mathematical functions: fabs, fmin, fmax, fsqrt, floor, ceil
- added new standard library functions: ftoa, ftoe, atof
- added functions for power management support and SPI
- added width specifier for the printf and sprintf functions
- added new #pragma savereg compiler directive
- enhanced the code optimizer for loops and comparisons
- enhanced the library functions for multiplication and division
- structures and unions can now be nested up to 16 levels
- added support for in-system programming using the MicroTronics ATCPU and Mega2000 boards
- enhanced the serial communication terminal

V1.0.0.5 Commercial Release

- enhanced the source file editor, added: undo, redo, mouse drag and drop, line numbers, color setup
- modified the mode how external RAM size is specified in the Project|Configuation|C Compiler menu
- enhanced the code optimizer
- revised the 90s2333.h, 90s4433.h, 90s4434.h, 90s8534.h, 90s8535.h files
- added Dallas Semiconductor 1 Wire protocol functions
- added Dallas Semiconductor DS1820 Temperature Sensor functions
- added Philips PCF8583 Real Time Clock functions
- added new functions: bcd2bin, bin2bcd
- added the @ operator for specifying the RAM storage address for global variables
- added the generation of a global variables memory map file by the compiler
- modified the ATmega103/603 programming
- added support for programming 3V devices
- updated the help file topic Working with the AVR Studio Debugger

V1.0.0.4 Commercial Release

- the names of project files are now stored in the .prj file as relative to the project directory
- increased the Terminal receive buffer size to 32K
- added inline assembly, ex: #asm("sei")
- added delay functions: delay_us, delay_ms
- improved the speed and size of the lsqrt function

- changed the programming of the FSTRT fuse bit for AT90S8535
- fixed some small bugs

V1.0.0.3 Commercial Release

- enhanced the code optimizer
- the AVR Chip programmer works also under Windows NT

V1.0.0.2 Commercial Release

- added struct and unions
- added global bit variables
- functions sqrt and lsqrt rewritten in assembly
- removed the csqrt function
- fixed some small bugs

V1.0.0.1 Commercial Release

- enhanced the code optimizer
- made the AVR Chip programmer compatible with the KANDA STK200 and STK300 development boards
- completed the functions library
- added support for: I²C bus, National Semiconductor LM75 temperature sensor
- modified the port bits allocation in lcd.h to simplify the circuit layout
- added new chip: ATmega103 with maximum 64K compiled binary size limit
- added a Serial Communication Terminal
- completed the Help file

V0.0.1.2 BETA

- added: bit level access to I/O registers
- added: support for ATtiny22, AT90S2333/4434, AT90S8534
- added: AVR Chip programmer
- added: Notes/Comments file attached to every project
- added in help file: explanations how to access from C functions written in assembly
- added: example c_asm.c how to access from C functions written in assembly
- corrected: lcd.h , csqrt.h , sqrt.h , lsqrt.h
- corrected: "missing } " error at the end of the source file
- corrected: Windows system errors when exceeding the BETA file size limit
- corrected: keyword highlighting in comments

V0.0.1.1 BETA

- added: arrays of pointers
- added: source level debugging in AVR Studio

- enhanced: the code optimizer
- added: history to the File menu
- corrected: the invisible error window problem
- small corrections to the help file

V0.0.1.0 BETA

- corrected: incorrect code generated for the != operator for 8 bit values
- corrected: relative jump out of range for large programs

V0.0.0.9 BETA

- first public release